JOINED

MILWAUKEE AREA ATARI USERS CLUB

MILATARI NEWSLETTER

Volume 2 Number 8

July 1983

Price $1.00

** NEXT MEETING **

SATURDAY, JULY 16st - Open at 2PM

ARMBRUSTER SCHOOL - GREENDALE

MEETING ADGENDA

1:00 PM Officers meeting

2:00 PM Technical session
Libraries open

3:00 PM Business meeting
Officers' reports
Old business
New business

4:00 PM **** EDUCATION ****
Joe Sanders - Education CIG chairman will head up an all education software demo.

5:00 PM Libraries open

** In This Issue **

Milwaukee Area ATARI Users Group

This newsletter is written and printed by members of the Milwaukee Area ATARI Users Group (MILATARI), an association of individuals with a common interest in using and programming ATARI computers. MILATARI is not affiliated with the ATARI company, nor any other commercial organizations.

All articles are written and donated by the membership. Opinions expressed in this publication are those of the individual author and do not necessarily represent, nor reflect, the opinions of MILATARI nor those of any other commercial or non-commercial organizations. Any article appearing in this newsletter may be reproducted, providing credit is given to the author and to MILATARI.

Write MILATARI Newsletter at P.O. Box 1191, Waukesha, WI 53187.

MEMBERSHIP INFORMATION

Membership is open to individuals and families who are interested in using and programming ATARI computers. The membership includes the subscription to this newsletter and access to the user's library. The membership fee is $15 per year for individual, $20 for family and $10 for associate. Contact Larry Leskovsek, Treas. at 547-0249 or write MILATARI, P.O. Box 1191, Waukesha, WI 53187 for more information.

MEETING INFORMATION

MILATARI meetings are held once monthly. This month the meeting will be held at the Armbruster School, 7000 Greenway, Greendale, WI. The meeting is held in the multi-purpose room. BASIC classes begin at 2:00 P.M. Technical sessions are also held a 2:00 P.M. The business session begins at 3:00 P.M. followed by demostrations. The library will be open before and after the business meeting.

MILATARI Officers:

President	Gary Nolan 353-9716
Vice-president	Chris Stieber 529-2663
Treasurer	Larry Leskovsek 547-0249
Secretary	Jim Comaris 353-3447
Eduction Chairperson	Linda Scott 466-2314
Cassette Librarian	Ron Friedel 354-1717
Disk Librarian	Steve Booth 367-8739
Publications Librarian	Karl Buschhaus 774-2576
Newsletter Editor	David Frazer 542-7242
Bulletin Board SYSOP	Bill Simotti 352-1790

- - - - - - - - - -

Technical support Group:

The following members have indicated a willingness to assist MILATARI members.

William Lawrence	1-968-3082 Programming
Don Wilcox	228-1650 Programming
Erik Hanson	252-3146 Prog/Tech
Gary Nolan	353-9716 Prog/Tech
Steve Booth	367-8739 Programming
Nick Liberski	786-8434 Prog/Tech

- - - - - - - - - -

MILATARI Bullentin Board:

The MILATARI Users Group maintains a 24 hr bulletin board service. The phone number is 352-2772.

PRESIDENT'S RAM

by Gary Nolan

WHERE'S THE ASPIRIN ???

Trying to assimilate everything from the CES could give anybody a headache. And even though we went through some of the new products at last months meeting, a lot of things were not mentioned. So here is a brief summery of products.

PRINTERS

EPSON has two new models, RX and FX. RX is intended to be a companion printer to the EPSON HX-20 computer. It has many MX features. The FX is the newest printer boasting 160cps, 12K ROM capacity and a claimed 128 different user-selectable type styles. It comes in 10" and 15" sizes.

ALPHACOM has a line of thermal printers with a difference. That's the connecting plug that allows you to plug directly in to the Atari serial port or other computers using RS232, Centronics parallel, IEEE 488 or to the VIC 20 and the 64. They come in 20,40 and 80 col. models. And all use the OLIVETTI print mechanisms. When used with the Atari interface module plug they will print the internal characters of the Atari computers.

STAR Micronics has three new printers. A $199 thermal model and two upgrades to the Gemini 10. STX-80 is a thermal printer with a speed of 60cps, two type sizes, standard ASCII plus 51 European chars., 64 block graphic chars. and bit image graphics. The Gemini 10X has 120cps, optional 4K or 8K buffers, 240x144 high res. graphics, Italics and the ability to download your own char. set to the printer. The Delta-10 printer is a 160cps model with 8K buffer standard and a 16K option and specs similar to the 10X.

The RITEMAN printer is small enough to fit into a briefcase, but has features that compare with full-sized models. It has 120cps speed, bit image graphics, Italics, tractor-pin-platen feed. While it only has two basic print sizes you can expand and enhance and double strike those sizes. It also supports the standard EPSON command codes.

For $250 you can get a Gorilla BANNANA printer. It's slow (50cps) and has only one basic print size (10cps) that can be expanded. Graphic resolution is 63x60, prints in one direction only and takes only tractor fed paper. But it's cheap.

PRINTER/PLOTTERS

Several new low cost printer/plotters were introduced at the show. The first we all know, is the ATARI 1020 four color, 40 col.. Standard print size is 10cps but has 64 sizes available.

Next comes the C.Itoh CX-4800. Features include four colors, 80 col., Centronics parallel, print speed of 8cps and plot speeds of up to 4.8"/sec.. You have tha ability to intermix printing and plotting on the same page. Plotting size is 7.6"x8" and will work with most micro's. List price is $695.

ROLAND's DXY 800 is an eight color model that comes with a parallel and serial interface. It plots at speeds of up to 7"/sec, is IBM compatible and will work with other micro's.

TYPEWRITERS

You say you need more than just a printer. Well there are some t/w's that can be used as computer printers also. They're usually more expensive than straight printers, but can serve a dual purpose. Most are Electronic Typewriters which means they have more capabilites than standard electric models. Here a but a few models.

Smith-Corona Memory Correct III Messenger has both parellel & serial input, changable print wheels and many other non-computer use features. Price is $575.
The Royal Alpha 2Ø15 prints at 14cps has a parellel interface and costs $7ØØ.
The Silver-Reed EX43 with the IF44 module will work as a computer printer. It has an average speed of 9cps and two print sizes on one wheel.
While not a true typewriter the COMREX CR-II is a daisy-wheel printer that you can add a keyboard to. It sells for $629 and the keyboard for $199. If this is the one I'm thinking of it has three type sizes 10-12-15cps built into one wheel.

OTHER STUFF

One printer related pieces of equipment that got our attention was The SPRINTER from the ALIEN GROUP. It features a 62k print buffer with video output and a keypad. The keypad allows you to scroll forward/backward through the text, jump forward/backward to selected points and to selectively print and erase any part of the file. It also allows you to review an 8Ø col printout on computers with 4Ø col displays. It will be helpful to assembly language programmers also in that you can get sorce code listings as fast as the assembler can generate it. Price is around $5ØØ.
I'll only comment on two of the hundreds of joysticks that were on display. Everything from foot operated models to some that you sit on. The Kraft joystick looked and felt good. If you get aggravated trying to make a diagonal move with some of other sticks check this one out. Very smooth response. The other one that stood out wasn't only a joystick an exercycle interface. This joystick/interface fits between the bike and the computer. It was hooked up to a 8ØØ and was running one of the raceing type programs, and the faster you peddled the faster you moved down the road. Now if they could hook up a fan to it, you could get some realism.
Monitors, graphic tablets, ect..ect.Everywhere you turn. The only thing bigger for a computer freak is the NCC. Gee, maybe the club could send someone (guess who?) to the next one.

SOFTWARE

Keeping track of the hardware is hard, keeping track of the software is impossible. There isn't enough room to review all the software but here's some of it.
Have you seen the movie Porky's? Well now they have a game based on the movie. How about games based on Dallas, Heathcliff, Bruce Lee, M*A*S*H and even (gimme a break) Heckle and Jeckle. Even Walt Disney is getting into the act. Games of every type. Space, sport, maze, monster, adventure, war and educational games.
One interesting program was Home Calc a sort of poor mans Visi-Calc. Intended for home use it'll run on a 16K cassette based or a 24K disk machine. Priced at $3Ø and $4Ø, it should be interesting. We could have one to demo at the next meeting.
But the newest and possibility the most intriguing development in software is the re-fillable cartridge. Not really re-fillable but reprogramable. The story goes something like this. You go to the software store and buy a cartridge, six months later you go bake and for a small fee they erase your program and load a new one into the EEPROM. Since most of the cost is the actual cartridge costs should come down.

We will have some literature at the next meeting on these and other products. Some was taken last time but there's still more. There will not be any from ATARI though. When I told them who I was and what I wanted it

for, I was asked if I would be willing to PAY for it. ADVERTISING LITERATURE! THE STUFF THEY GIVE AWAY ! BY THE TON!! PAY FOR IT!!!!! I thanked them , but declined thier kind (gag) offer.
I wonder if anybody at ATARI really reads the newsletters we send to them. I'd really like to hear them explain that. PAY FOR THE STUFF!!!!!!

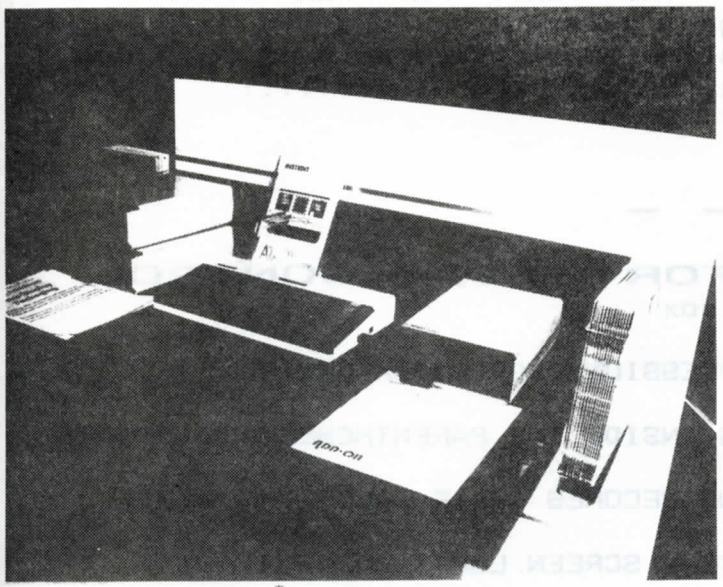
Be there on the 16th for my next tirade.

— — — — —

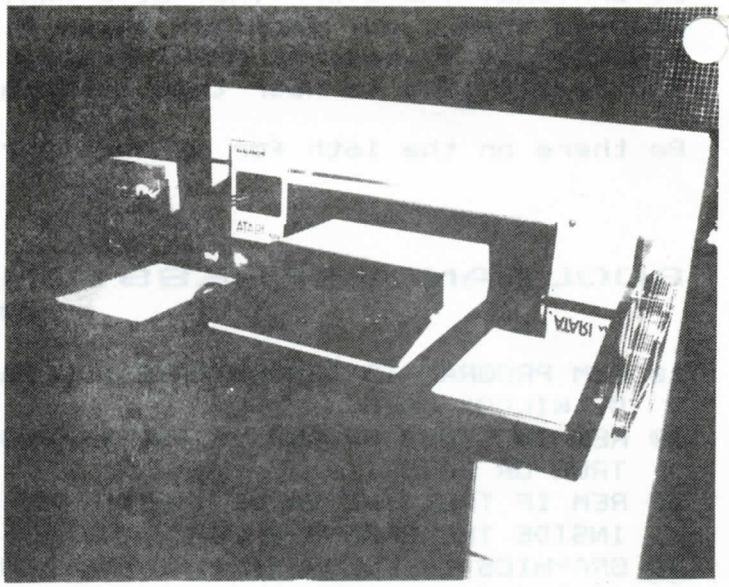
BOOLEAN EXPRESSIONS FOR STICK CONTROL

by Don Wilcox

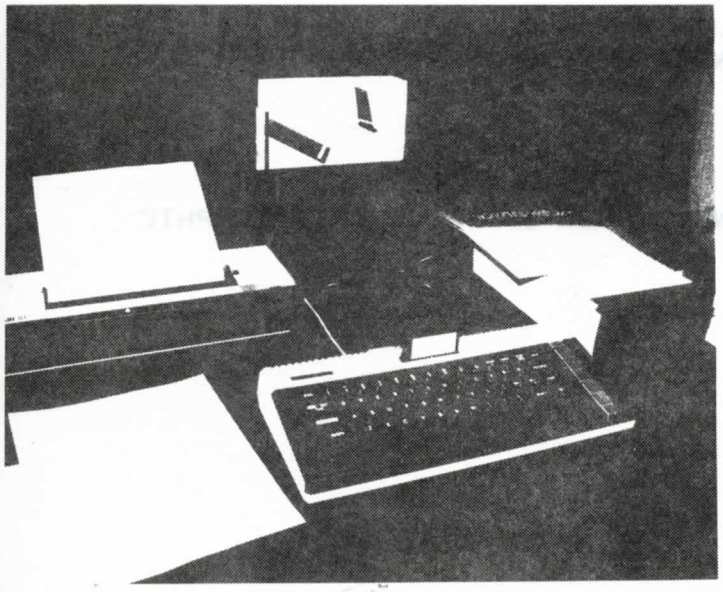
```
10 REM PROGRAM TO DEMONSTRATE BOOLEAN EXPRESSIONS FOR STICK CONTROL.
   D. WILCOX  6-83
20 REM IN LINES 80 AND 90 THE EXPRESSIONS INSIDE THE PARENTHESES ARE EITHER
   TRUE OR FALSE.
30 REM IF TRUE THE VALUE INSIDE THE PARENS BECOMES 1. IF FALSE THE VALUE
   INSIDE THE PARENS BECOMES 0.
40 GRAPHICS 19:COLOR 1:X=10:Y=10:REM INITIAL SCREEN LOCATION OF PIXEL
50 PLOT X,Y
60 S=STICK(0):T=STRIG(0):IF S=15 AND T=1 THEN 60:REM STICK MOVED OR TRIGGER
   PRESSED? IF NOT, STAY HERE
70 IF STRIG(0)=0 THEN ? #6;"}":GOTO 50:REM PRESS TRIGGER TO CLEAR SCREEN
80 X=X+(S<8)*(X<39)-(S>8 AND S<13)*(X>1):REM HORIZONTAL MOVEMENT
90 Y=Y-(Y>1)*(INT(S/2)=S/2)+(Y<20)*(INT(S/4)*4=S-1):GOTO 50:REM VERTICAL
   MOVES
100 REM IF YOU CHANGE THE GRAPHICS MODE, BE SURE TO CHANGE THE BOUNDARY
    CHECKING (X<39 AND Y<20)
110 REM STICK MOVEMENTS FOLLOW A MATHEMATICAL PATTERN
120 REM LEFT *     (S>8 AND S<13)
130 REM RIGHT *    S<8
140 REM UP *       S IS ALWAYS AN EVEN NUMBER
150 REM DOWN *     ALWAYS LEAVES A REMAINDER OF 1 IF S/4
160 REM LEFT       BOUNDARY X>1
170 REM RIGHT      BOUNDARY X<39
180 REM TOP        BOUNDARY Y>1
190 REM BOTTOM     BOUNDARY Y<20
200 REM REMEMBER THESE BOUNDARY CONDITIONS CHANGE IF YOU CHANGE GRAPHIC
    MODES.
```



2 ATARI 1050 Disk Drives 800XL CPU CP/M module



ATARI 1450XL with built-in disk drive



ATARI's answer to ADAM
ATARI 600XL w/ATARIWRITER and
ATARI 1027 letter quality
printer

Gary and friend check out
Mattel's AQUARIUS

PRODUCT UPDATE

ATARI HOME COMPUTER SYSTEM

ATARI® BASIC Reference Manual Update

This product update contains a number of corrections and additions to the *ATARI BASIC Reference Manual.*

Page 1. This definition is missing from the **TERMINOLOGY** list:

Floating Point Number: A number containing an integer part, a decimal point, and a fractional part. The total number of significant digits in a floating point number, excluding the exponent, may be either nine or ten. This depends on whether the exponent is an even or odd multiple of 10.

Page 6. This information pertains to the **ARITHMETIC OPERATORS** subtraction and exponentiation:

Note: Avoid negating zero, as this will produce an invalid number. For example, if you type

```
PRINT -0
```

the result will be

```
-0E-<8
```

Note: Since the algorithm used to generate exponents (^) is only an approximation, you cannot obtain integer results with it—for example, 2^2=3.99999996. To correct this, use the following technique:

```
X=2^2
PRINT INT (X+.5)
4
```

Page 7. This Note regards the use of the **LOGICAL OPERATORS:**

Note: Avoid using the statement PRINT A=NOT B, as the results are not predictable. Essentially, any PRINT statement with a NOT operator will be unpredictable.

Page 13. This Note is in reference to **SCREEN EDITING:**

Note: Large amounts of editing may lock up the system. It's recommended that programs under development be stored to cassette or diskette periodically (every 30 or 40 edits) with the SAVE or CSAVE command.

Page 20. This Note regards **ON/GOSUB** statements:

Note: If an ON/GOSUB expression evaluates to a number greater than the number of subroutine entries, then a POP statement will be necessary to clear the stack (see POP command, Section 4).

Page 22. Further information on **RESTORE (RES.):**

The RESTORE statement will not generate an error if the line number referenced does not exist. Instead it will RESTORE to the next larger line number in the program. Care should be taken to update RESTORE statements when renumbering a BASIC program.

Page 25. Some additional information on using the **INPUT (I.)** statement:

When executing an INPUT from the screen, avoid moving the cursor away from and then back to the same line; otherwise, the wrong data may be input. Specifically, the INPUT prompt will be included in the INPUT string.

If a string of 128-255 characters is INPUT, then RAM locations 1536-1664 will be overwritten. This area is normally reserved for storage of programs or data. (See the *ATARI Tech Reference Notes.*)

To INPUT strings of more than 127 characters, use the GET command and store the values into a string (see Section 5, OPEN/CLOSE and PUT/GET commands).

Note: The maximum number of characters that can be INPUT from the screen is 120. The maximum for other devices is 255.

Note: Make sure that every INPUT statement has a variable after it; otherwise, unpredictable results may occur.

Page 26. This regards the use of the **LOAD (LO.)** command:

Note: If a program is loaded that is too large for the available memory space, it may give unpredictable results without an error message.

This Note should follow the **LPRINT (LP.)** command description:

Note: An LPRINT command with a semicolon at the end will cause the following LPRINT statement to print on the next 40-column tab. A 40-column printer will move to the next line in such a case. To use the semicolon effectively, use the OPEN statement for the printer, then write to the printer with a PRINT statement (see OPEN/CLOSE and PRINT commands, Section 5).

Page 27. This information pertains to the filespec definition:

Note: Be sure to include the closing quotation marks on a filespec parameter, especially when putting multiple statements on one line. For example,

```
OPEN #1, 4, 0, "D:TEST":STOP
```

will work, but

```
OPEN #1, 4, 0, "D:TEST:STOP
```

will not function correctly.

Page 28. This is an addition to the **POINT (P.)** section:

Note: To update a file, you must open it with a 12 in aexp1.

In the last paragraph under **PRINT (PR. or ?)**, the first sentence should read:

A comma tabs every 10 spaces.

The following sentence should conclude the final paragraph on **PRINT (PR. or ?)**:

However, if the last character to be printed (as in a string with quotation marks) is a CTRL R or CTRL U, then the next PRINT will begin at the end of the current line.

This note should then conclude this section on **PRINT (PR. or ?)**:

Note: In rare circumstances data printed to a diskette may have part of the BASIC program embedded in it. If this occurs, retry the operation.

This Note regards the **PUT (PU.)/GET (GE.)** section:

Note: In certain circumstances the GET function may modify other variables within the program. To avoid this, PRINT any number to the screen between each GET.

Page 30. Here is a corrected version of the table—note in particular the correction on cmdno 32:

cmdno	OPERATION	EXAMPLE
3	OPEN	Same as BASIC OPEN
12	CLOSE	Same as BASIC CLOSE
13	STATUS REQUEST	Same as BASIC STATUS
17	DRAW LINE	Same as BASIC DRAWTO
18	FILL	See Section 9
32	RENAME	XIO 32,#1,0,0,"D:TEMP,CAROL"
33	DELETE	XIO 33,#1,0,0,"D:TEMP.BAS"
35	LOCK FILE	XIO 35,#1,0,0,"D:TEMP.BAS"
36	UNLOCK FILE	XIO 36,#1,0,0,"D:TEMP.BAS"
37	POINT	X10 37,#1,A,B
38	NOTE	X10 38,#1,X,Y
254	FORMAT	XIO 254,#1,0,0,"D2:"

Page 33. The last sentence in the paragraph about the **CLOG** function should read:

CLOG(0) through CLOG(1) are inaccurate and should not be used.

Page 34. The last sentence in the paragraph about the **LOG** function should read:

LOG(0) through LOG(1) are inaccurate and should not be used.

Page 38. The last line in the first paragraph should read:

was stored there previously.

Page 39. The first sentence should read:

Upon execution, the screen displays THE SQUARE ROOT OF 10000 IS 100.

In the second paragraph, the last line should read:

number 1000000000.

This is additional information on the **VAL** function:

Only the numeric field will be translated, while the text will be ignored. For example:

VAL(''5SUM'')=5

This information pertains to **String Concatenation:**

Note: BASIC cannot move strings of 256-character multiples correctly. String lengths should be checked; if any string contains a multiple of 256 characters, add or subtract one character from the amount to be moved.

In *Figure 7-6,* the correct result of the program on the left is:

BCD#

Page 42. Some additional information on using the **DIM (DI.)** statement:

Make sure that the DIM statement does not contain a space between the string or array name and the left parenthesis of the dimensioned amount; otherwise, the following will happen—

DIM L (10) becomes DIM L10)

—and this variable can no longer be referenced.

Note: The command COM is identical to DIM and may be used in its place.

Page 43. This is an additional Note for the **DIM (DI.)** section:

Note: Due to a discrepancy in boundary checking, arrays of up to 32766 by 32766 in size can be dimensioned. The programmer should size the array ahead of time to ensure that there is no ''virtual'' storage space.

Additional information on using the **CLR** command:

The second sentence in the last paragraph, beginning ''It also clears ...,'' should be deleted.

The CLR command will not initialize the values in strings and arrays.

Page 45. Here is a corrected version of **TABLE 9.1:**

TABLE 9.1—TABLE OF MODES AND SCREEN FORMATS

Gr. Mode	Mode Type	SCREEN FORMAT Horiz. (Columns)	Vert. (Rows) Split Screen	Vert. (Rows) Full Screen	Number Of Color Registers	RAM Required (Bytes) Split Screen	RAM Required (Bytes) Full Screen
0	TEXT	40	—	24	1½	—	992
1	TEXT	20	20	24	5	674	672
2	TEXT	20	10	12	5	424	420
3	GRAPHICS	40	20	24	4	434	432
4	GRAPHICS	80	40	48	2	694	696
5	GRAPHICS	80	40	48	4	1174	1176
6	GRAPHICS	160	80	96	2	2174	2184
7	GRAPHICS	160	80	96	4	4190	4200
8	GRAPHICS	320	160	192	1½	8112	8138

Page 49. The last sentence under **PLOT (PL.)** should read:

"The range of points begins at 0 and extends...."

Page 50.

In **TABLE 9.3**, the color PURPLE should be inserted after PINK in the first column, and the number 5 should be inserted after 4 in the second column.

Page 51. The sentence directly under **TABLE 9.4** should read:

"DEFAULT" occurs if no SETCOLOR statement is used.

Page 53. Here is a corrected version of **TABLE 9.5:**

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
		0	Color data	—
LIGHT BLUE	Mode 0 and	1	actually	Character luminance
	all text		determines	(same color as background)
DARK BLUE	windows	2	character to	Background
		3	be printed.	—
BLACK		4		Border
ORANGE		0	Color data	Character
LIGHT GREEN	Modes 1	1	actually determines	Character
DARK BLUE	and	2	character to be	Character
RED	2	3	printed.	Character
BLACK	(text modes)	4		Background, border
ORANGE		0	1	Graphics point
LIGHT GREEN	Modes 3, 5,	1	2	Graphics point
DARK BLUE	and 7	2	3	Graphics point
	(four-color	—	—	—
BLACK	modes)	4	0	Graphics point (background default), border
ORANGE	Modes 4	0	1	Graphics point
	and 6	—	—	—
	(two-color	—	—	—
	modes)	—	—	—
BLACK		4	0	Graphics point (background default), border
		—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background)
DARK BLUE	Mode 8	2	0	Graphics point (background default)
	(1 color, 2 luminances)	—	—	
BLACK		4	—	Border

Page 54. In *Figure 9-4,* line 80 should read:

80 XIO 18, #6, 12, 0, "S:"

Page 55. This information pertains to **TABLE 9.6:**

In Column 1, # 14, a period, not a bar, shows on the screen.

In Column 3, #'s 92-95 should show a superscripted circled 1 next to their characters.

Page 56. Here is a corrected version of **TABLE 9.7:**

TABLE 9.7—CHARACTER/COLOR ASSIGNMENT

		Column 1 Conversion	Column 2 Conversion	Column 3 Conversion	Column 4 Conversion
MODE 0	[2]SETCOLOR 2	#+32	#+32	#-64	NONE
		POKE 756,224		POKE 756,226	
MODE 1	SETCOLOR 0	#+32	#+32	#-32	#-32
OR	SETCOLOR 1	NONE	#+64	#-64	NONE
MODE 2	SETCOLOR 2	#+160	#+160	#+96	#-96
	SETCOLOR 3	#+128	#+192	#+64	#+128

[2] Luminance controlled by SETCOLOR 1, 0, LUM

Page 58. The last paragraph should read as follows:

Note that the DATA statement in line 90 ends with 256, which is outside of the designated range. The 256 is.....

In **TABLE 10.1:**

The PITCH VALUE of 193 should have a musical note of "E," not "D."

Page 63. The last line in item 9 should read:

precedence will save a few bytes.

Page 67. In *Figure 11-2,* line 0260 under Data should be:

#2

Page E-1.

The right parentheses are missing after the word "CONSTANT" in Atari Functions of Inverse Cosine, Inverse Secant, and Inverse Cosecant.

Page H-7. Line 160 in the program should read:

160 IF K=125 OR K=155 THEN 180

Page H-8. Line 50 in the program should read:

50 PLOT 0,0:DRAWTO 159, DR

Page 117.

Following COM, "(see DIM)" should be deleted and replaced with "A-1."

Page 118.

Under "Input/Output Devices," Line Printer should be followed by "(P:)," not "(L:)."

Page 119.

"NOTE, 26" is missing from the listing.

Page 120.

INSIDE BACK COVER here is the corrected table:

"STATUS, 29" is missing from the sublisting under "Statement" and also from the regular listing.

MODE, SETCOLOR, COLOR TABLE

Default Colors	Mode or Condition	SETCOLOR (aexp1) Color Register No.	Color (aexp)	DESCRIPTION AND COMMENTS
		0	Color data	—
LIGHT BLUE	Mode 0 and	1	actually	Character luminance
	all text		determines	(same color as background)
DARK BLUE	windows	2	character to	Background
		3	be printed.	—
BLACK		4		Border
ORANGE		0	Color data	Character
LIGHT GREEN	Modes 1	1	actually determines	Character
DARK BLUE	and	2	character to be	Character
RED	2	3	printed.	Character
BLACK	(text modes)	4		Background, border
ORANGE		0	1	Graphics point
LIGHT GREEN	Modes 3, 5,	1	2	Graphics point
DARK BLUE	and 7	2	3	Graphics point
	(four-color	—	—	—
BLACK	modes)	4	0	Graphics point (background default), border
ORANGE	Modes 4	0	1	Graphics point
	and 6	—	—	—
	(two-color	—	—	—
	modes)	—	—	—
BLACK		4	0	Graphics point (background default), border
		—	—	—
LIGHT BLUE		1	1	Graphics point luminance (same color as background)
DARK BLUE	Mode 8	2	0	Graphics point (background default)
	(1 color,	—	—	
BLACK	2 luminances)	4	—	Border

C061038 REV. A

ATARI COLOR GRAPHICS

Examples and discussions of the use of Color Graphics on the ATARI 400/800 Home Computer System

1) Four-color Modes
2) Five-color Text Modes
3) Screenful of Hearts

Items 4 thru 10 will be in the August issue:

4) Etch-a-Sketch
5) Circlez
6) Fill a Shape
7) GTIA Graphics Modes
8) Swirl
9) Race
10) Bumper

Information provided by:

ATARI INC.
CONSUMER PRODUCT SERVICE
PRODUCT SUPPORT GROUP

DEMOPAC #4

— — — — —

ATARI COLOR GRAPHICS
Four-Color Modes 3,5, and 7
JB 5/82

In four-color modes, four of the system's five color registers are available for use. There are three such modes, GRAPHICS 3,5, and 7. All three are high resolution, or map modes. This means that individual pixels on the screen are turned on and off, instead of being grouped together into characters. The three modes have different resolutions, or pixel sizes, but the color characteristics are the same.

One of the four available registers controls the background color, so you get three foreground colors. The COLOR statement is used to select one of the four color registers, until another COLOR statement is executed. Each register contains a default hue, as follows:

COLOR 1 selects register 0, which contains the color orange.
COLOR 2 selects register 1, which contains the color green.
COLOR 3 selects register 2, which contains the color blue.
COLOR 0 selects register 4, which contains the color black.

In all map modes, COLOR 0 selects the background register. In modes 3,5, and 7 this is register 4, which contains black.

You must use a COLOR statement to select a foreground register before doing any PLOTs, or your plotted points will not show up. If no COLOR statement is encountered, the register defaults to the background, register 4.

The SETCOLOR statement is optional. It is used to change the color information in a register. The SETCOLOR numbers correspond directly to the register numbers. If you select COLOR 1, and do not wish to use orange, you would change the color in that register with SETCOLOR 0. The statement is:

COLOR 1: SETCOLOR 0, hue, luminence

COLOR 1 selects register 0, and the contents of register 0 are controlled by SETCOLOR 0. When the SETCOLOR statement is executed, everything which you have drawn using COLOR 1 will change to the new hue and luminence.

To erase a point or line, simply draw over it using the background color, COLOR 0. To erase everything in one color, use SETCOLOR to change that register to the background color. Black is hue 0 and luminence 0. A table of hue numbers can be found on page 50 of the BASIC Reference Manual. For luminence, select an even number from 0 to 14, 0 is the darkest and 14 the brightest luminence.

NOTE: There is an error in the Hue Table on page 50. Hue 5, Purple, is missing. Please make the change in your manual.

ATARI COLOR GRAPHICS
Five-Color Text Modes
JB 5/82

Graphics modes 1 and 2 are the Text-Graphics modes. All five of the system's color registers are available for use, one as background, and four as characters colors. Mode 2 characters are larger, but color is handled the same way in both modes.

There are two ways to put graphics-characters on the screen. The character can be placed at an x-y coordinate with the COLOR and PLOT statements, or it can be PRINTed through channel #6:

```
PRINT#6;"A"

or

COLOR 10:PLOT 5,5
```

The PLOT command allows you to specify where the character is to go, but not which one it is. The COLOR statement is used in these modes to select a character. The character number is taken from the Internal Character Set chart on page 55 of the BASIC Reference Manual.

SELECTING A REGISTER WITH PRINT#6

The color of a character is determined by which register is used to draw it, but the COLOR statement does not select a register in text modes, as it does in map modes. Selecting a register is a little complicated. Let's take the case of PRINT#6 first.

The statement PRINT#6;"A" prints an orange A. Orange is the default hue in register 0. Registers 1-3 are selected with the lower-case and logo (inverse video) keys. The lower-case key selects register 1, which contains green. The logo key selects register 2, which contains blue. Both keys at once select register 3, which contains red.

To print four letter A's in four different colors, use the following statement:

```
PRINT#6;"AaAa"(the final characters
      are in inverse video)
```

Remember, in these modes, you cannot get the upper-case and lower-case character set at the same time. When you use the lower-case key, you are selecting a color register, not a lower-case letter. To get a lower-case 'a' you have to POKE 756,226 in order to use the lower-case character set. You cannot get lower-case and upper-case letters together on the same screen.

No matter what method you are using to put your characters on the screen, you can change the color contained in the register by using he SETCOLOR statement. SETCOLOR numbers are the same as register numbers. Any one of the 128 colors can be placed in any of the five color registers.

SELECTING A REGISTER WITH COLOR/PLOT

To select a register for a COLOR/PLOT, you must use the charts on pages 55 and 56 of the BASIC Reference Manual. Find the character in the Internal Character Set chart on page 55. If it is in column 1 or 2 it is in the upper-case set. If it is in column 3 or 4, it is in the lower-case set, and you must POKE 756,226 to use it.

Once you have found your character number, look at the Character/Color Assignment chart on page 56. Registers are referred to by SETCOLOR number. To assign a register to the character, you must add the conversion numbert to the character number.

If you choose a character from column 1, and wish to assign register 1, the table tells you that no conversion is necessary. When you put the character number in the COLOR statement and PLOT it, the register defaults to 1. Since register 1 contains green, the character is green.

The statement COLOR 10:PLOT 5,5 puts a green asterisk on the screen. The internal character number for a asterisk is 10, which is in column one. With no conversion, register 1 is selected, so it comes out green.

To select another register for your character, add the offset given in the conversion table to the character number, and put this new number in your COLOR statement. To PLOT an asterisk using register 3, use the statement:

```
COLOR 10+128:PLOT 5,5
```

+128 is the conversion for column 1 and register 3. Since register 3 defaults to red, the new star is red.

Any character in column 2 requires conversion. The letter A is character number 33, in column 2. If you want to PLOT an A, you must select COLOR 33+conversion. If you simply PLOT with COLOR 33, you get an orange exclamation point (character 1+32).

NOTE: There is an error in the conversion chart, which you should correct in your manual. Under Conversion 1, SETCOLOR 0, you should change -$32 to $+32. To convert a column 1 character to register 0, add an offset of 32 to the character number.

The registers contain the default colors mentioned, but these colors can be changed with the SETCOLOR statement, as in all other modes.

When you change a hue using SETCOLOR, everything written with that register changes to the new hue. If you want characters in separate colors, you must use separate color registers. A character can be erased by plotting or printing over it in the background color, using the corresponding SETCOLOR command.

GRAPHICS 1 and 2
Use Lower Case Without a Screenful of Hearts
JB 4/82

In Text-Grapics modes 1 and 2, the upper-case and lower-case character sets can only be used separately. Because the lower-case character for a space contains a heart, calling up the lower-case set results in the screen filling up with hearts. The following article, reprinted from The Atari Connection #1, explains in detail why that happens, and provides a program to correct the situation. The program moves the character set from ROM down into RAM, and then redefines the heart-character as a space-character.

When you use the character redefining program, you must include the program itself as part of your initialization. When you then call lower-case, use something like the following routine:

```
200 POKE 756,CHBAS+2:REM instead of the usual 226
210 POSITION 5,5:PRINT #6;"LOWER CASE"
220 POSITIION 3,8:PRINT #6;"WITHOUT HEARTS"
```

Other characters can also be redefined using the same program. Simply change the character number (CHAR) in line 110, and the DATA in line 130.

Refining the character set takes time, and reduces the amount of RAM available for your program. There is an alternative

which is much simpler. You can simply erase the hearts by making them the same color as the background. The only problem with this is that you lose one of your color registers: you can use only three character colors, instead of the usual four. If this is OK in your program, use something like the following routine:

```
10 GRAPHICS 1
20 POKE 756,226:REM call lower-case
30 SETCOLOR 0,0,0:REM set register 0 to black
40 FOR I=1 TO 5
50 READ X
60 COLOR X+64:REM add 64 to get another color
70 PLOT I,5
80 NEXT I
90 DATA 33,52,33,50,41
```

Try taking out line 30, and the hearts come back. The data and the offset for the color change are taken form the charts on pages 55 and 56 of the BASIC Reference Manual. The COLOR statement is used in modes 1 and 2 to determine which character is to be plotted.

THE CASE OF THE UNWANTED HEARTS

If you program in graphics mode 1 or 2 and use lower case characters, little hearts appear on the screen where spaces should be. The way the computer makes letters results in these hearts.

Your computer doesn't know what printed characters look like. It must interpret them using an internal table called a "character set". Graphics modes 1 and 2 have two different character sets, each with 64 different characters. One character set contains all the upper case characters (capital letters) and punctuation marks. The other set contains all the lower case letters and graphics characters.

Each key on the keyboard matches a position in the character set. The first postion of all character sets represents the space bar. Thus, the computer looks in the first position of the character set whenever the space bar is pressed to see what to put on the screen.

The first position of the upper case character set contains a space, and the first position of the lower case character set contains a heart shape. Therefore, if you use upper case characters, you will see a space on the screen whenever you press the space bar, but if you use lower case characters, you will see a heart.

The computer clears the graphics screen by filling it with the internal character for the space bar. Thus, if you use the lower case character set, the screen will become filled with hearts instead of blank spaces.

Normally, you get the upper case characters, but you can use lower case if you use the command POKE 756,226. Two methods exist to avoid getting the unwanted hearts with this character set.

The first and easiest way is to set the hearts to the screen's background color by using the command SETCOLOR 0,0,0. This method, however, uses up one of your colors.

A better way, but it requires a little more work, is to change the character set so the first lower case character becomes a blank space rather than a heart. Before you try to change the character set, an understanding of its arrangement in the computer's memory would be helpful.

A character set is simply a picture of what the letters will look like when they are printed on the screen. If you look closely at the printing on the screen, you will see that the letters actually consist of little dots. The computer represents each dot as a single bit of information in its memory. Each memory location can store eight bits (one byte) of information, or eight dots. A bit can either be 1 or 0. If a bit is set to 1, it represents a dot on the screen. If it is set to 0, it represents a place where no dot shows, or a blank space. The computer represents each character as an 8X8 arrangement of dots.

Therefore, the computer uses eight locations in memory for each character. For example, the heart character looks like the following in your computer's memory:

```
00000000
00110110
01111111
01111111
00111110
00011100
00001000
00000000
```

See how the 1's and 0's form a heart?

Each row of 1's and 0's can also be read as a number. Page 55 of the BASIC Reference Manual displays the entire character set and corresponding numbers for your ATARI Personal Computers. Each box represents eight locations in the computer's memory and contains the image of one character. The heart is character 64 in the set. A space is character 0 in the set. The space would be represented by a block of 0's in memory. (Do you remember why)

Your computer's character set is stored in read only memory (ROM), but as a result of the versatility of the ANTIC chip, we can move the character set into the read/write memory (RAM) and the characters will still be displayed correctly. Once we have the set in RAM, we can modify the characters to our heart's content.

To avoid the unwanted hearts in lower case you need to move the character set into RAM and change the heart to a space, which the program listed below will do. From then on, a simple POKE statement will switch you from upper case to lower case characters.

The first step required is to determine if adequate memory space exists for the program to change the heart to a space. This program requires about 4600 bytes of memory, so the first line in the program listing, Line 10, checks to see if 4600 bytes of memory are free.

If adequate RAM exists, you need to make room above RAMTOP for the character set. The character set occupies 1K of memory. The computer's Display List occupies a little less than 1/2K. Since information is accessed from memory in 256 Byte or 1/4K pages, you must move RAMTOP down 4 pages to allow 1K of space (1/4K per page X 4 pages=1K). The current value of RAMTOP is stored in the RAM location 106. Line 20 looks into the memory location for RAMTOP (106) and sees what its present value is. Line 30 subtracts 4 pages, moving RAMTOP down the needed 1K.

Before you move your character set, you need to let the computer know you moved RAMTOP so it can save its Display List. You do that with a graphics command. Line 40 sets up the computer to use graphics mode 1. If you move the character set before doing a graphics command, you might write over the Display List and ruin your display.

Now, you're ready to move the character set from ROM to the last 1K of RAM, just before the old RAMTOP. We'll call the beginning of the character set CHBAS. Since 1K equals 4 pages of memory (1/4K per page X 4 pages=1K), Line 50 tells the computer where CHBAS, your new character set, should be located in RAM. Line 60 tells the computer where the CHBAS should begin. (Remember, one page equals 256 bytes.)

```
10 IF FRE(0)<4600 THEN PRINT "NOGO":END
20 RAMTOP=PEEK(106)
30 POKE 106,RAMTOP-4
40 GRAPHICS 1
50 CHBAS=RAMTOP-4
60 ADDR=CHBAS*256
70 FOR X=0 TO 1023
80 POKE ADDR+X,PEEK(57344+X)
90 NEXT X
110 CHAR=64
120 POS=ADDR+(CHAR*8)
130 DATA 0,0,0,0,0,0,0,0
140 FOR X=0 TO 7
150 READ A
160 POKE (POS+X),A
170 NEXT X
180 STOP
```

In ROM, the old character set began at location 57344. To move the character set so its RAM location begins at ADDR, the actual location of the character set, you need Lines 70, 80 and 90. Line 80 looks into the ROM memory location and then moves the character set to its new location.

All you need to do now is change the heart to a space. Remember, the heart is character number 64, each character requires eight locations in memory, and the character st begins at ADDR. As a result, the beginning location of the heart charcter is ADDR+64*8. Lines 110 and 120 read character 64 from memory and put it into the correct location.

To replace the hearts with spaces, recall that eight values are required to represent a character in memory. For the space, these values are easy to determine...all the values are 0! Line 130 defines character 64 to be represented as all 0's or to be a space.

Once the character has been defined with the DATA command, another FOR-NEXT loop (Lines 140-170) will load it into the character set.

If you use this routine in a program, type in your other lines of code next. If not, end the program with a STOP

command (Line 180). Otherwise, the computer will automatically close the graphics display area and prevent you from being able to print in the graphics area.

After you have entered this routine, if you want the upper case character set, use the command POKE 756,CHBAS. For the lower case character set, use the command POKE 756,CHBAS+2. The hearts will be gone!

COMPUTERESE DICTIONARY

Antic Chip--A microprocessor that enables the computer to display graphic images on the television screen.

Bit--Smallest amount of information your personal computer can work with; must be either a 1 or a 0.

Byte--Amount of information your computer requires to represent a character (letter, number or symbol); one byte consists of eight bits.

Character Set--Table in the computer's memory which contains a picture of what the letters, numbers and symbols will look like when they are printed on the screen.

Display List--Machine language program for the ANTIC chip.

FRE--BASIc command which determines amount of memory available.

PEEK--A BASIC command that tells the computer to look into a specific location in the computer's memory and see what is stored there.

POKE--A BASIC command that tells the computer to put a new number into a specific location in the computer's memory.

RAM--Acronym for Random Access Memory; computer memory for temporary storage of information; the computer can both read information stored here and write new information to be stored here.

RAMTOP--Top of memory; the number put in this location is the number of 256-byte pages of memory space available for RAMTOP.

ROM--Acronym for Read Only Memory; computer memory for permanently stored information; the computer can read information stored here but cannot write new information to be stored here.

MILATARI Newsletter
David Frazer, Editor
P.O. Box 1191
Waukesha, WI 53187-1191

U.S. POSTAGE PAID
BULK RATE
PERMIT NO. 201
Waukesha, WI 53186

Address Correction Requested
Forwarding and Return Postage Guaranteed